ART Event Mixing: do we want it in LArSoft?

LArSoft General Mtg
Eric Church (Yale)15-Jan-2013

Event Mixing

- Run many different physics samples, and overlay them.

- Mu2e/CD's Rob Kutschke has done the work as appropriate to mu2e.

- I will approach this as a uBooNE collaborator

Candidate overlay samples

- CRY cosmic rays that rain down on uBooNE. We would run once and create a look-up cosmics library and overlay them, rather than walk them through G4 all the time.

- GENIE rock events, in which the topVolume is set to something like volWorld. Same idea. Would overlay them on GENIE detector events. However, overlaying here could mean just accumulating simb::MCTruth's and pushing everything through LArG4 together the way we do it now for cosmics+GENIE. No real EventMixing bookkeeping required. That's because here the bottleneck is at generation.

How Sarah/Brian run Cosmics now

- They run a lar job that does the following:

- runs a 3-frame CRY cosmic event module

- runs a 1-frame GENIE event with say BNB flux

- runs LArG4 with all above 4-vectors input, and walks them through G4. This does all the trackID book-keeping nicely

- Then DetSim+CalWire+Recon

event times through detsim (no optical sim)

- `/uboone/data/users/uboonepro/gen/S2012.12.17/`**`prodgenie_bnb_nu_t0_uboone`**`/6834794_*/lar.out:` **123 sec/evt**

- `/uboone/data/users/uboonepro/gen/S2012.12.17/`**`prodcosmics_uboone`**`/6813564_*/lar.out:` **300 sec/evt**

- Above is only 1 frame. That needs to be tripled, which from an early result from Sarah doesn't add much to the running time. Still, we could save 70% of our cpu/wall-clock time by pre-running cosmic muons and overlaying them later. Haven't fully characterized time consumption/savings.

- I don't have a top volume == worldVol for comparison, but one can imagine that the need to run GENIE evts with this setting will lead to very few detector interactions, and so something like event mixing is mandatory to get any reasonable number of rock muons at all overlaid.

Real EventMixing is more complicated

- Separate lar jobs are run to produce unique ART cosmics/rock/signal .root files.

- The master lar job is then launched which runs the art::EventMixing module that sucks in the cosmics/rock/signal files, and event by event merges each output object from each upstream ART module.

- In so doing objects are "flattened," meaning, e.g., the individual truth PtrVectors from each parallel module are combined to give just one.

- Along the way trackIDs, e.g., must be made unique: parents, daughters, self

- Similarly sim::IDEs must be made to understand these new unique trackIDs for the backtracker's sake, e.g.

Important Question of where to mix

- mu2e has 4+ background files: decay-in-orbit-muons, protons, neutrons, ... They walk through G4 but don't hit their sensitive detectors. They properly merge just before detectors are hit by examining the particle trajectories, then accumulate the response on to sensitive detectors to make G4Hits. One can imagine sums that cross the pedestal response to give hits which taken individually would otherwise not.

- We, however, drift our electrons inside G4. So, it's too late to merge prior to wire charge formation.

- I'd propose that we ignore pedestal issues/non-linearities and sum charge at each tick per wire after LArG4. Otherwise there's no point to the cosmics overlay time-savings. (There still remains a point for the pre-G4 rock muon overlay.)

Now ...doing it.

- The mu2e module/class is available: http://cdcvs0.fnal.gov/cgi-bin/public-cvs/cvsweb-public.cgi/mu2e/Offline/EventMixing/src/MixMCEvents_module.cc and Rob is very helpful.

- There are summary methods that can be used as-is. There are simplifications we can make: no need for re-sorting mu2e's art::Ptr's since LArSoft doesn't write such things out -- at least not in LArG4 and prior. We just want to take 3 cosmics frames from one file and overlay on the 1 GENIE frame from another. I *think* all art::Assns are preserved during flattening.

- That's the good news.

- However, each set of objects to be flattened requires new methods in the EventMixing class that *we* have to write, that drill down into the individual objects and make all information unique, as mentioned in the G4 case discussed on slide 6. There's a callback scheme between ART-EventMixing that I don't claim to understand yet ...

Don't know what's next.

- It's probably realistically a person-week of work to do EventMixing

- Do we want to pursue it?

- Perhaps the place to start is a Rock+Signal mixing where only N poisson rock simb::MCTruth's need merging into GENIE detector event.